HW1 Big Data Platform

Submitters

Tzachi Hakmon 312412927
Michal Deutch 206094989
Ofri Kutchinsky 313611360

Twitter data

Due to API usage limitations, we implemented our twitter insight service to word with data taken from twitter Kagle data set. Having said that, we Implemented the service in generic, extendable and portable manner such that most of its component agnostic to the data source and will work exactly the same if tomorrow we'll decide to extend it and to plugged in to another data source.

Twitter insights.

Our service suggests an API with two main insightful endpoints:

1. Get top K most trended topics for a given period (user input is start date and end date).
2. Get yearly trend of single topic for a given period (user input is the year and the topic).

For the bonus part and to demonstrate one of nginx.ingress advantage over simple cluster load balancer, we exposed another endpoint that "post" tweets to twitter. The implementation there is poor but intended to demonstrate the ability to expose many different APIs' of twitter based services we could host on maintain on our cluster.

K8S Cluster structure.

Our K8S Cluster is designed around a microservices architecture, following a 3-layer model for streamlined operations. The architecture includes:

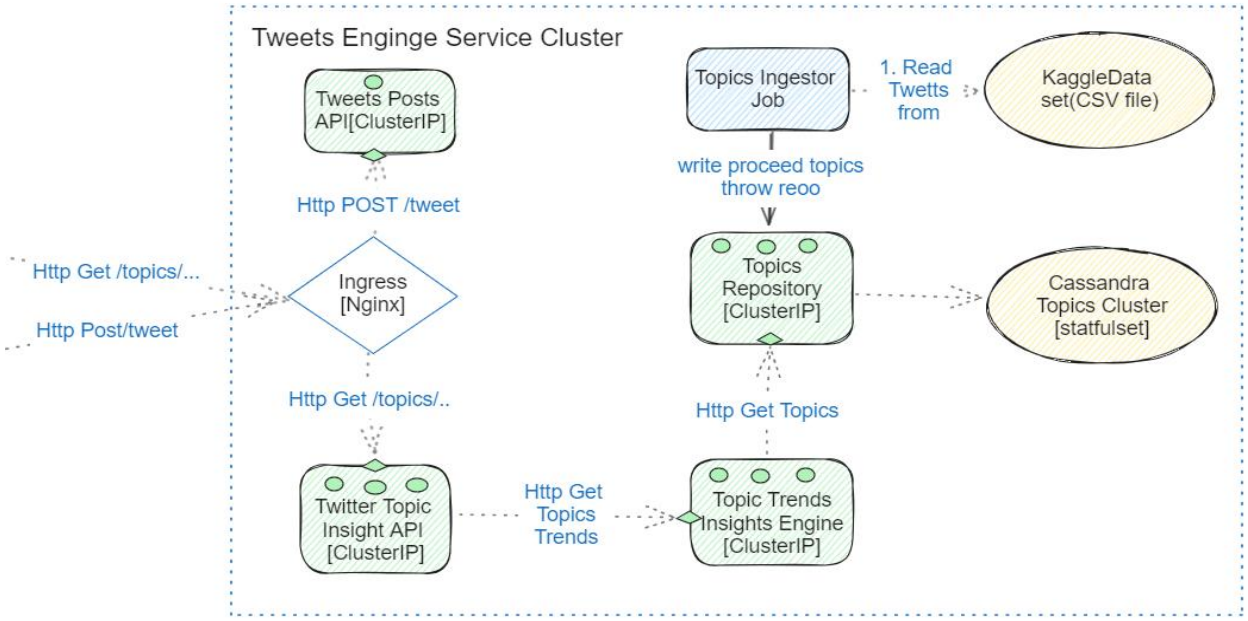
1. Public API - Serves as the user interface, handling requests, validating inputs, and communicating with the business logic layer. This layer is prepared for future enhancements like authorization and local caching. It primarily returns JSON objects but is flexible for other formats like HTML.
2. TrendsEnginge – Dedicated to processing and calculating topic trends based on user parameters. This layer handles intensive computations and its topics trends described further in a challenges section.
3. TopicsRepository – Manages data interactions, notably with Twitter topics, and encapsulate the data source from other service components. This layer integrates with a Cassandra DB within our K8S cluster.

In addition, we have three more k8s components in our cluster:

1. Topics ingestor job – Executes at cluster initialization, reading tweets, extracting topics using NLP to identify entities, and populating the Cassandra DB topics tables via the TopicsRepository.

2. Cassandra DB with one topic key space of two tables of topics. We decided to create two tables of topics from the same data because our business need. We designed two tables within a single keyspace to address distinct business requirements, optimizing query efficiency by tailoring partition and clustering keys. The `topics_by_time_table` utilizes `(year, month)` as its partition key to cluster data by time, enabling fast retrieval of topics within specific months, enhancing performance for trending topic analysis. Conversely, the `topics_trends_table` employs `topic` as its partition key, with temporal columns as clustering keys, streamlining the extraction of popularity trends for individual topics over time, thereby leveraging Cassandra's strengths for targeted data access.

3. Nginx Ingress controller – Acts as the gateway to our cluster, managing HTTP routing for our public APIs and ensuring efficient service delivery.



K8S Cluster configuration

Our architecture comprises three microservices, each with its Deployment and Service files, using ClusterIP for internal cluster communication. The Trends-Engine service is allocated more resources to manage its intensive tasks, preventing it from becoming a bottleneck. We utilize Horizontal Pod Autoscaler for dynamic scaling based on workload, with tailored settings for each service reflecting their processing demands. Config-Maps enhance our microservices' flexibility across environments. Additionally, we employ a Kubernetes Job for batch tasks, a StatefulSet for our Cassandra DB to ensure data persistence, and an Nginx ingress controller for routing in our API gateway.

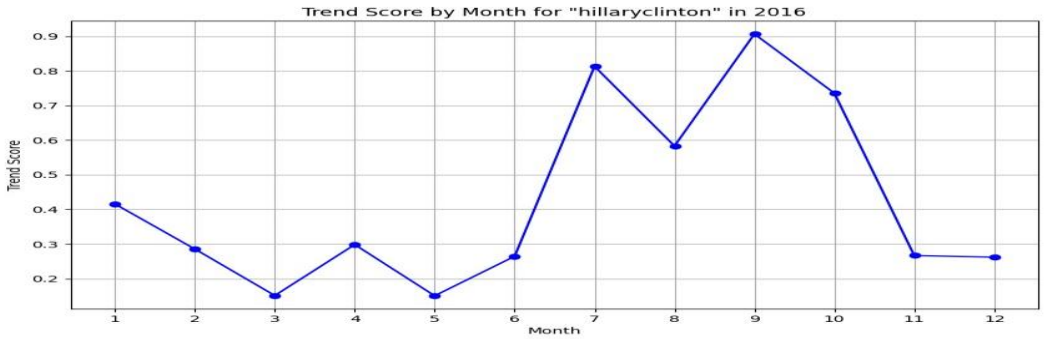
Key Insights derived from data.

To visualize results We created two python scripts with http clients that call our service endpoint. Script names: ***topic_trend_client_and_visualization_script***, and ***top_k_topic_words_cloud_client__and_visualization_script*** . Reviewer could adjust script to test the API with different arguments. We used WordCloud and matplotlib for visualization.
All the scripts outputs could be found under “visualization_results_samples\top25-visoalization” and “visualization_results_samples\topic_trends_over_the_year”

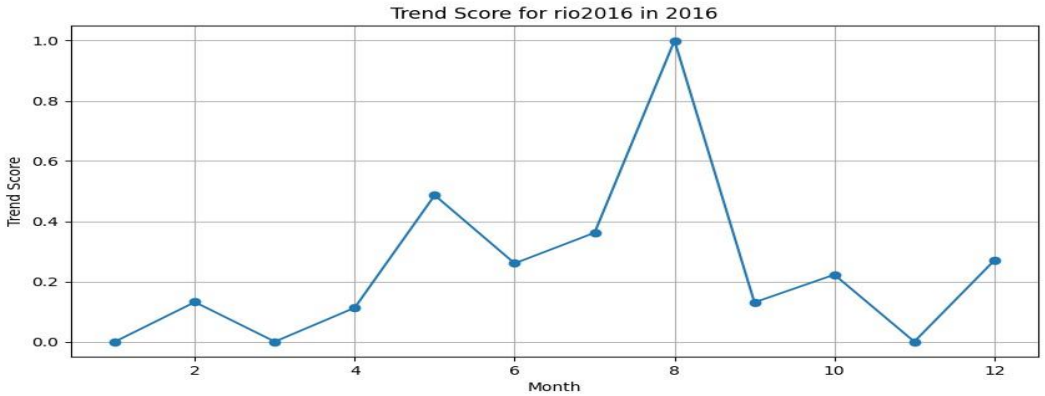
Interesting insights:

Single topic trend over the year.

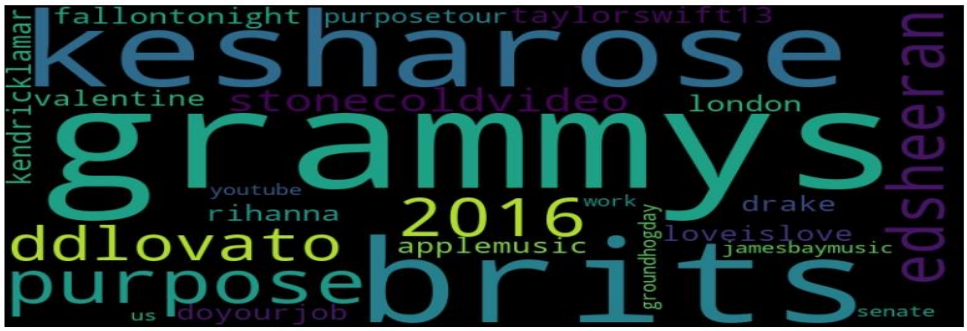
‘HillaryClinton’ trend score before and after US 2016 president elections that was took place on November:



rio2016 climbing and get to its peak during the Olympic games in RIO.



February 2016 top 25 trended topics are almost all related to American music industry as the 58th Annual Grammy Awards took place on 16/2/2016



Among most trended topics we can find the artists: Kesha(kesharose), Demi Lovato(ddlovato), Kendrick Lammar, Ed Sheeran, Rihana, Taylor Swift, Drake.

Challenges

1. **Calculating topic trend score.**
   To calculate topic trend score per period we used 4 measurements from them we created 4 sub score. each score component—share, like, frequency (number of tweets), and author frequency scores (number of distinct authors)—is calculated by comparing the topic's metrics against the highest values observed across all topics.
   To calculate the trend score for topics over time, we initially assigned equal weights to four metrics. However, we adjusted these weights to counteract biases—such as celebrities influencing tweet frequency disproportionately. For example, after observing skewed results due to prolific tweeters like Justin Bieber, we decreased the weight for like frequency and increased it for unique author count to ensure a more balanced and representative trend score.
2. **Efficient and precise topic extraction from tweets**
   To enhance topic identification, we initially utilized a library called YAKE, which employs statistical methods to pinpoint key phrases and terms. However, the output was suboptimal, including numerous irrelevant words that didn't meet our requirements. Seeking improvement, we turned to the Spacy library, renowned for its robustness and efficiency in performing various Natural Language Processing (NLP) tasks such as tokenization, part-of-speech tagging, named entity recognition (NER), among others. Utilizing NER, we effectively extracted significant subjects and entities from tweet content for topic extraction.
   Despite the library's effectiveness, further refinement of the results was necessary. We implemented a text cleaning mechanism to eliminate hashtags and mentions and converted all text to lowercase to avoid duplicates. Moreover, we introduced a filtering process using a static list of words to be excluded from Spacy's output, predominantly time-related terms like days, months, and periods, which we preferred not to consider as topics. This exclusion list is customizable and managed via our configuration map, allowing for tailored and precise topic identification.